Final_Exam

Daniel Briseno Servin

12/18/2020

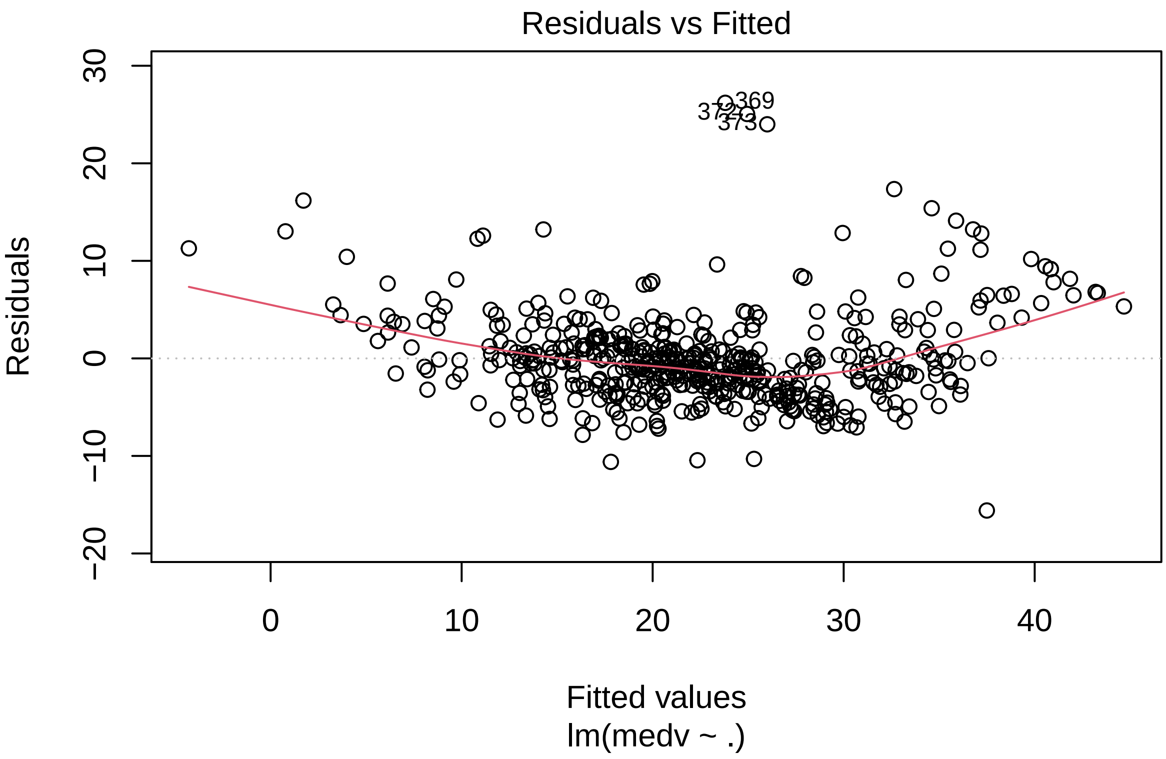
Problem 4

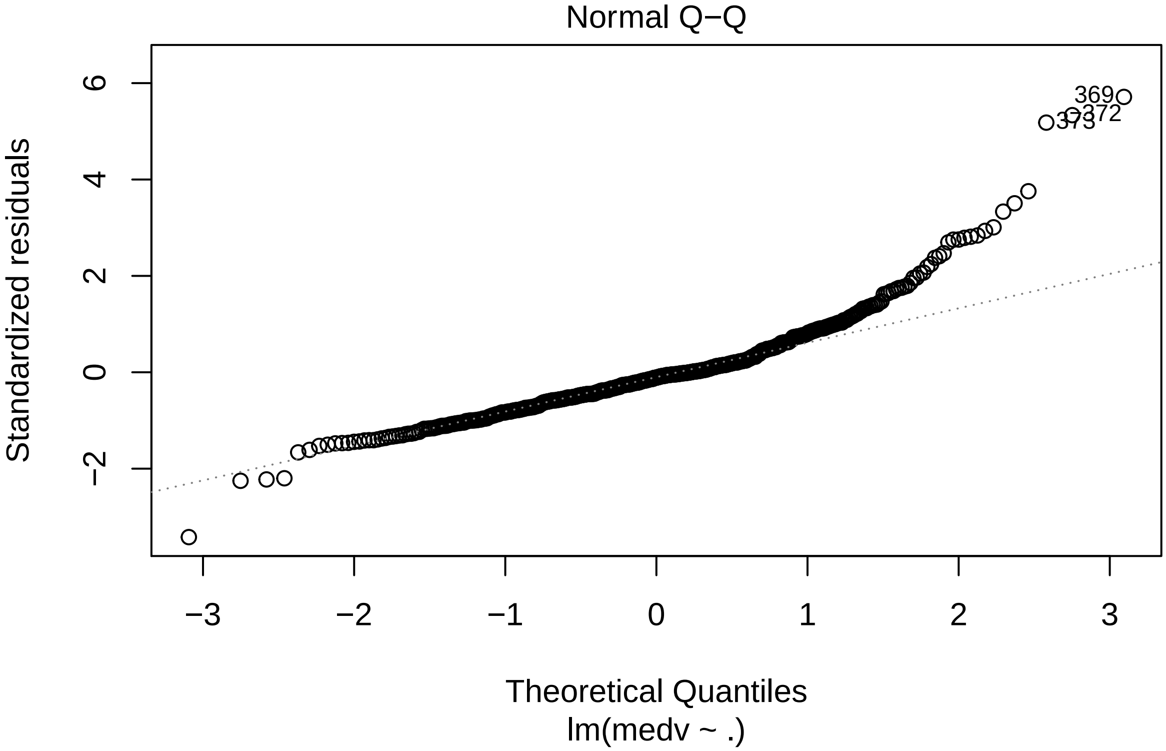
First the base-regression model

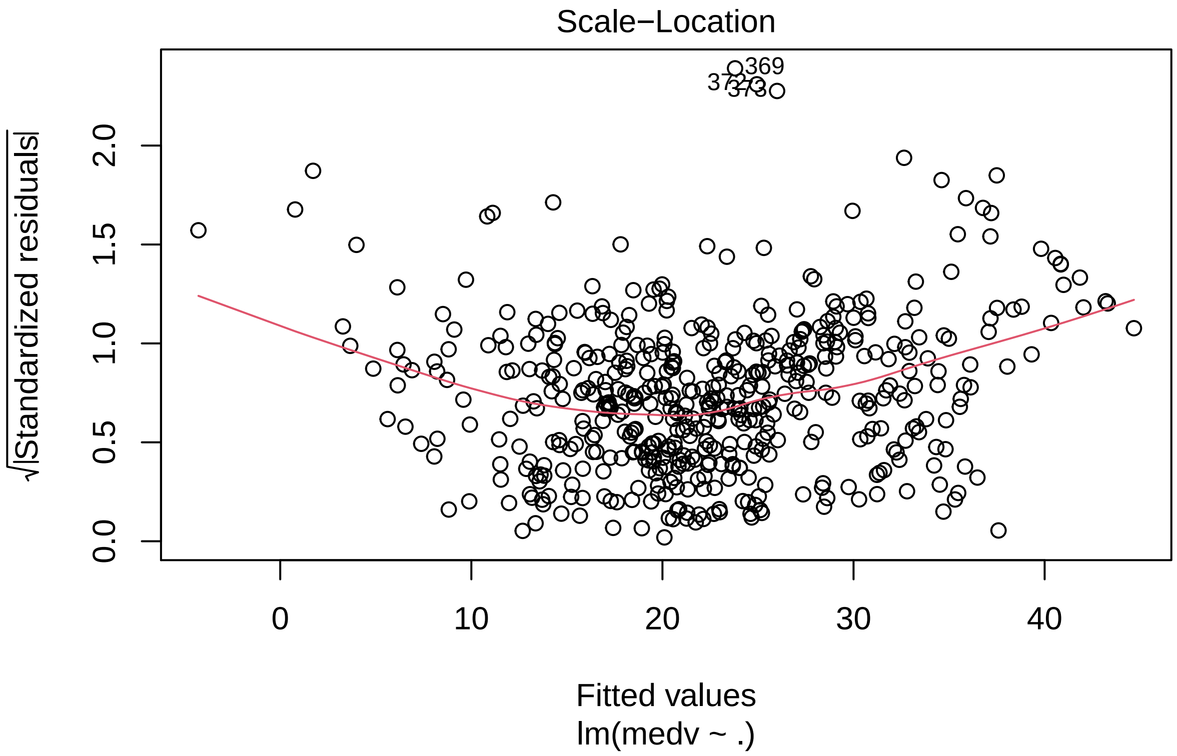
```
data(BostonHousing)
d <- BostonHousing
d <- na.omit(d)

# base regression model
base <- lm(medv~., data=d)
summary(base)
```

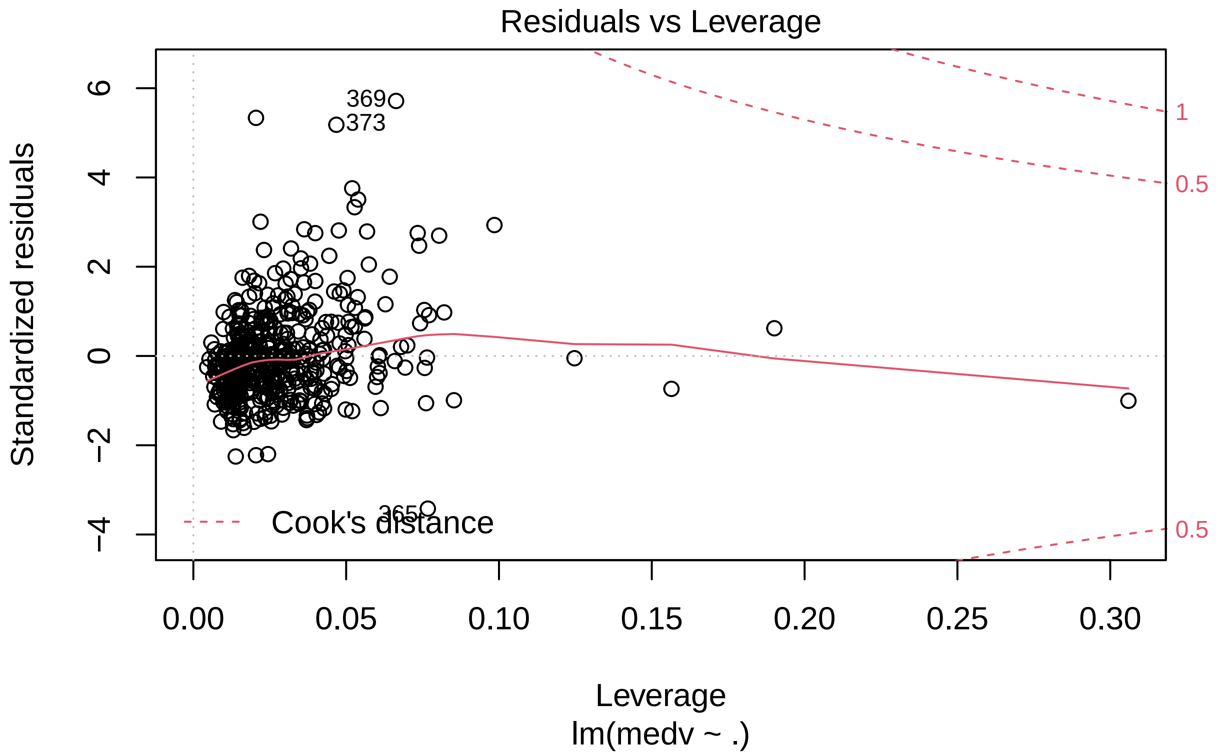
```
##
## Call:
## lm(formula = medv ~ ., data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas1        2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## b            9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```r
plot(base)
```


Residuals vs Fitted
Residuals
30
20
10
0
−10
−20
0
10
20
30
40
Fitted values
lm(medv ~ .)
369
372
373


Normal Q−Q
Standardized residuals
6
4
2
0
−2
−3
−2
−1
0
1
2
3
369
372
373
Theoretical Quantiles
lm(medv ~ .)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
10
20
30
40
Fitted values
lm(medv ~ .)
369
372
373

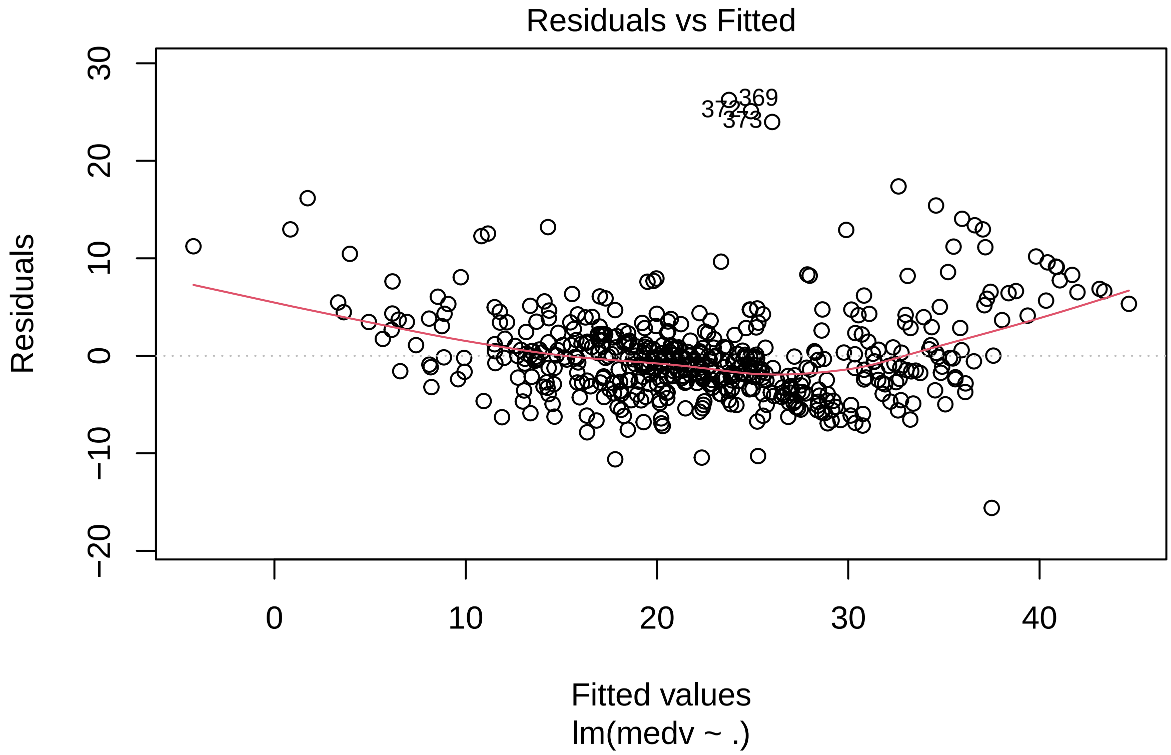


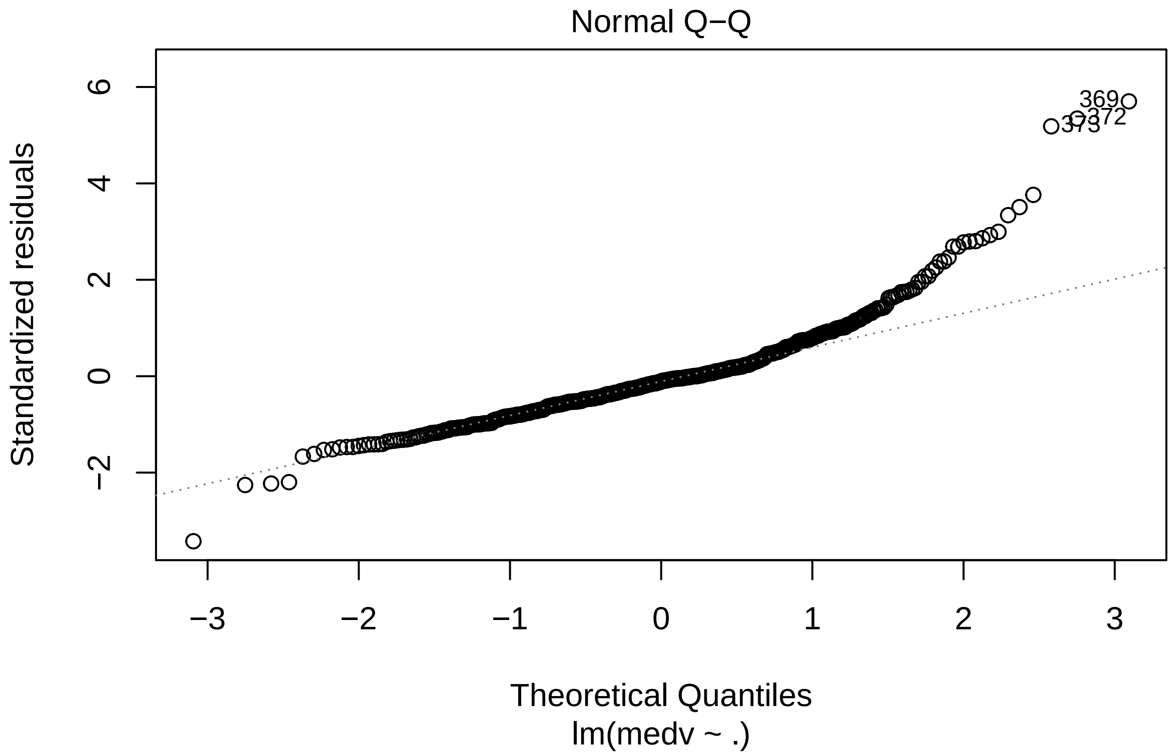
Base model with only stat. sig. variables

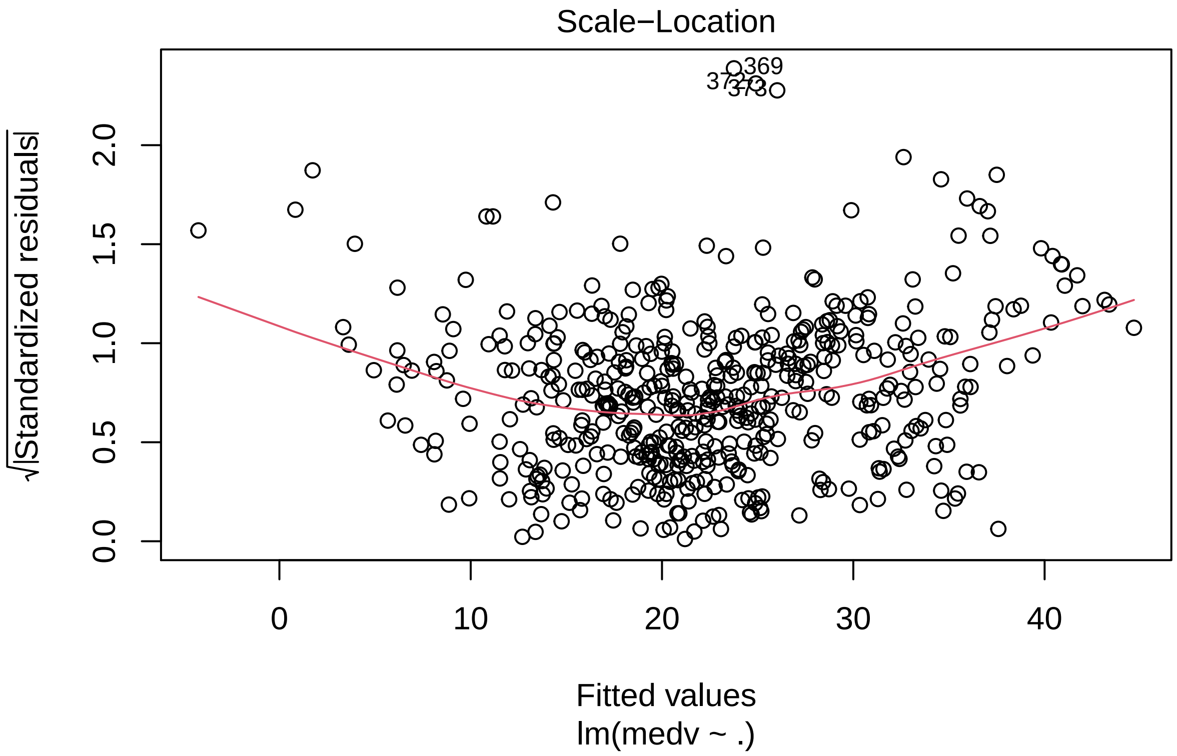
```
d_sig_base <- data.frame(d) %>% dplyr::select(-indus) %>% dplyr::select(-age)
base_sig <- lm(medv ~., data=d_sig_base)
summary(base_sig)
```

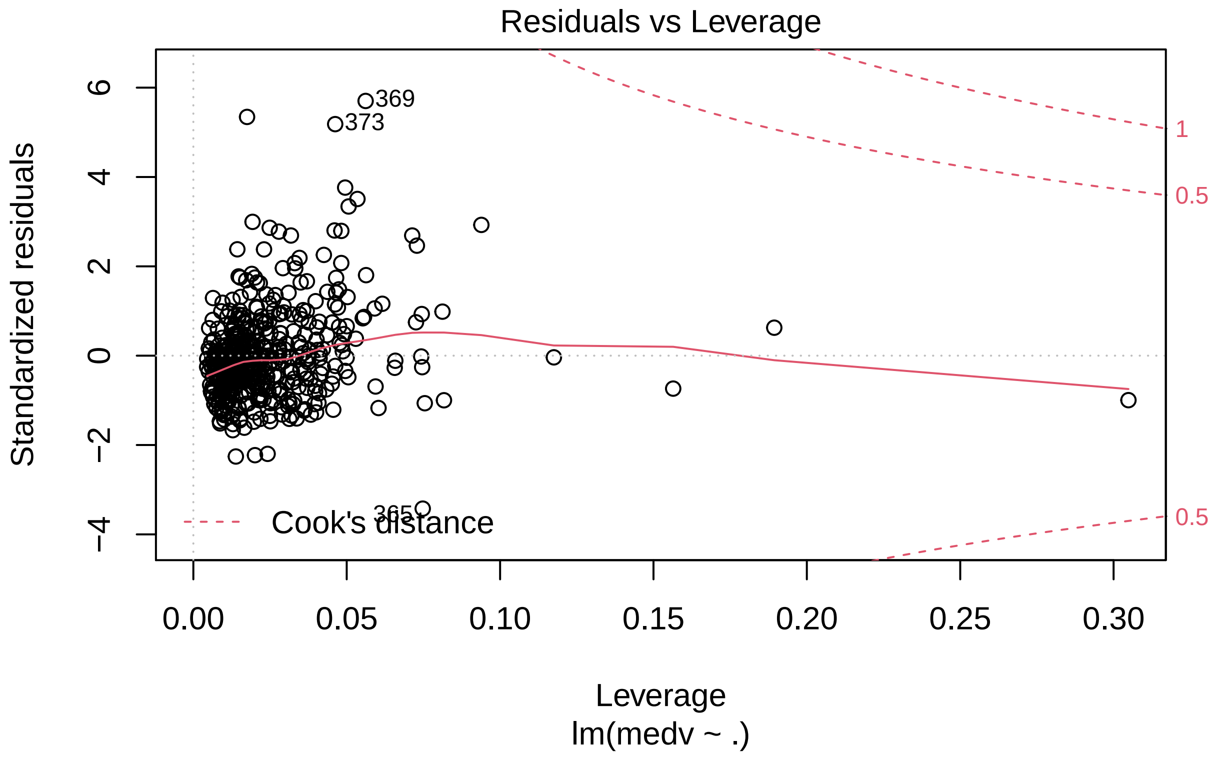
```
##
## Call:
## lm(formula = medv ~ ., data = d_sig_base)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5984  -2.7386  -0.5046   1.7273  26.2373
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
## crim         -0.108413   0.032779  -3.307 0.001010 **
## zn            0.045845   0.013523   3.390 0.000754 ***
## chas1         2.718716   0.854240   3.183 0.001551 **
## nox         -17.376023   3.535243  -4.915 1.21e-06 ***
## rm            3.801579   0.406316   9.356  < 2e-16 ***
## dis          -1.492711   0.185731  -8.037 6.84e-15 ***
## rad           0.299608   0.063402   4.726 3.00e-06 ***
## tax          -0.011778   0.003372  -3.493 0.000521 ***
## ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
## b             0.009291   0.002674   3.475 0.000557 ***
## lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.736 on 494 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7348
## F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

```r
plot(base_sig)
```




Normal Q−Q
Standardized residuals
6
4
2
0
−2
−3
−2
−1
0
1
2
3
369
372
373
Theoretical Quantiles
lm(medv ~ .)


Scale−Location
369
372
373
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
10
20
30
40
Fitted values
lm(medv ~ .)



Step -AIC Model

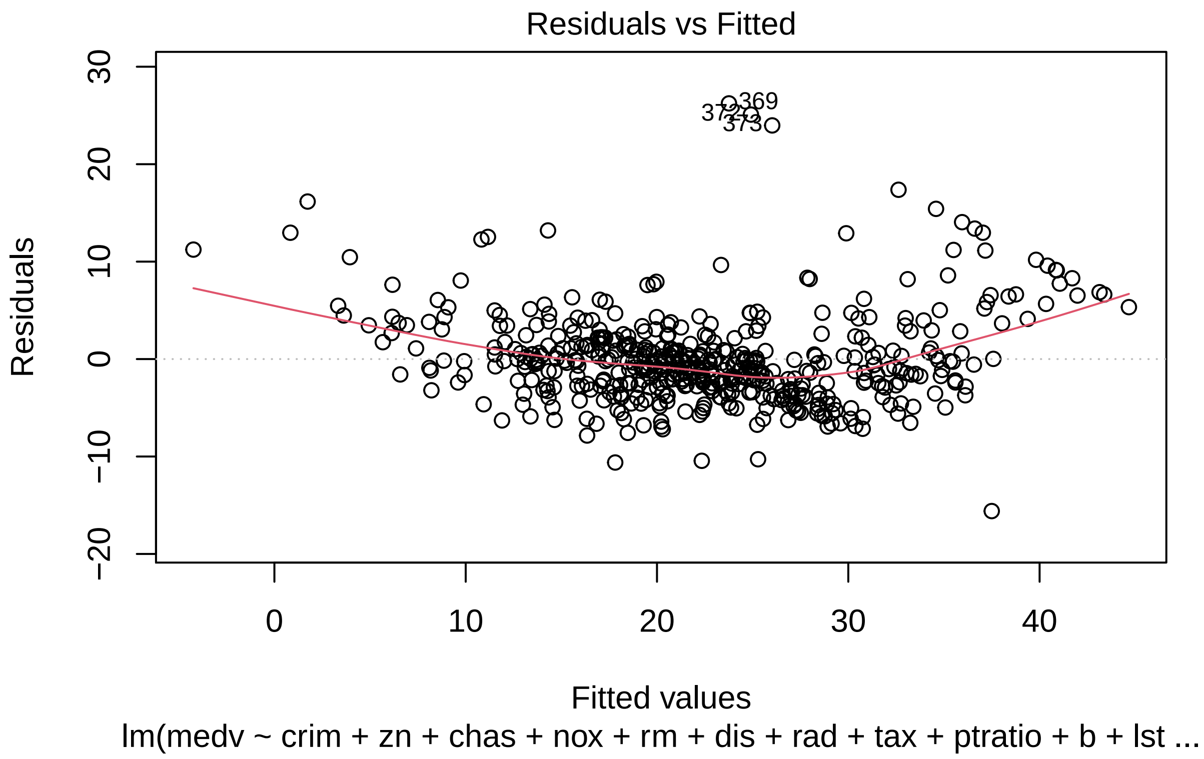
```
final <- stepAIC(base)
```

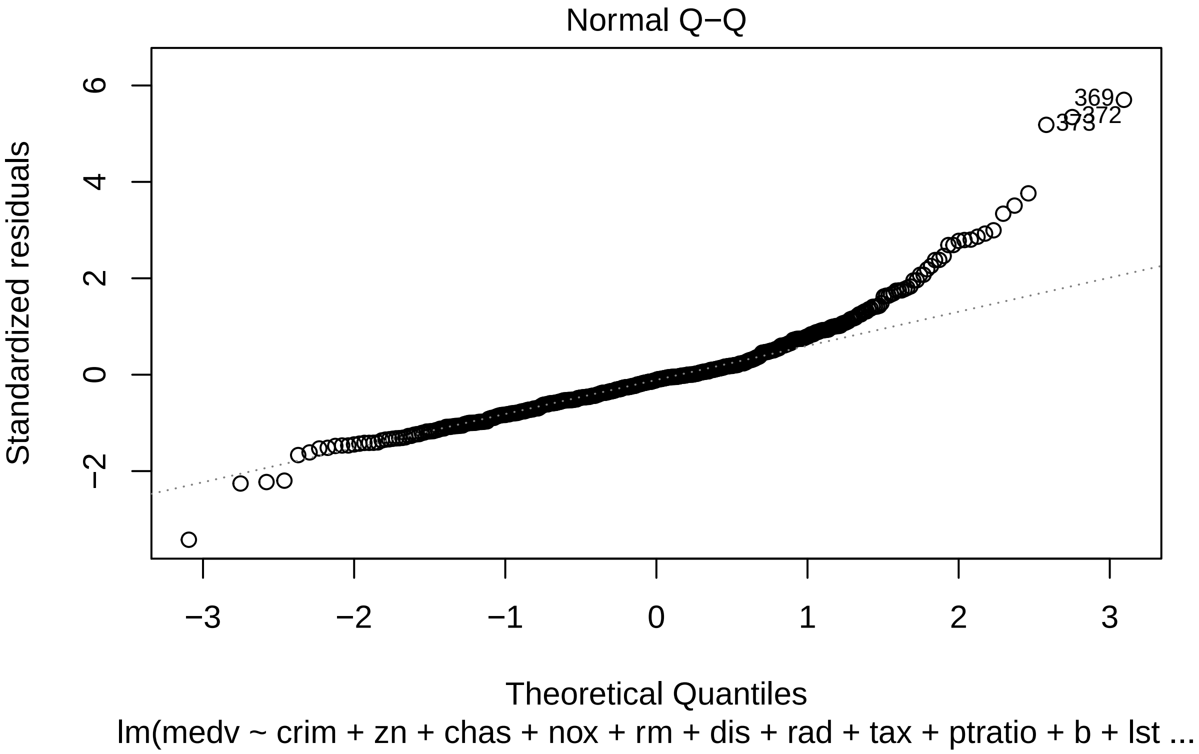
```
## Start:  AIC=1589.64
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + 
##     tax + ptratio + b + lstat
## 
##           Df Sum of Sq   RSS    AIC
## - age      1      0.06 11079 1587.7
## - indus    1      2.52 11081 1587.8
## <none>                 11079 1589.6
## - chas     1    218.97 11298 1597.5
## - tax      1    242.26 11321 1598.6
## - crim     1    243.22 11322 1598.6
## - zn       1    257.49 11336 1599.3
## - b        1    270.63 11349 1599.8
## - rad      1    479.15 11558 1609.1
## - nox      1    487.16 11566 1609.4
## - ptratio  1   1194.23 12273 1639.4
## - dis      1   1232.41 12311 1641.0
## - rm       1   1871.32 12950 1666.6
## - lstat    1   2410.84 13490 1687.3
## 
## Step:  AIC=1587.65
## medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax +
##     ptratio + b + lstat
##
##           Df Sum of Sq   RSS    AIC
## - indus    1      2.52 11081 1585.8
## <none>                 11079 1587.7
## - chas     1    219.91 11299 1595.6
## - tax      1    242.24 11321 1596.6
## - crim     1    243.20 11322 1596.6
## - zn       1    260.32 11339 1597.4
## - b        1    272.26 11351 1597.9
## - rad      1    481.09 11560 1607.2
## - nox      1    520.87 11600 1608.9
## - ptratio  1   1200.23 12279 1637.7
## - dis      1   1352.26 12431 1643.9
## - rm       1   1959.55 13038 1668.0
## - lstat    1   2718.88 13798 1696.7
##
## Step:  AIC=1585.76
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     b + lstat
##
##           Df Sum of Sq   RSS    AIC
## <none>                 11081 1585.8
## - chas     1    227.21 11309 1594.0
## - crim     1    245.37 11327 1594.8
## - zn       1    257.82 11339 1595.4
## - b        1    270.82 11352 1596.0
## - tax      1    273.62 11355 1596.1
## - rad      1    500.92 11582 1606.1
## - nox      1    541.91 11623 1607.9
## - ptratio  1   1206.45 12288 1636.0
## - dis      1   1448.94 12530 1645.9
## - rm       1   1963.66 13045 1666.3
## - lstat    1   2723.48 13805 1695.0
```

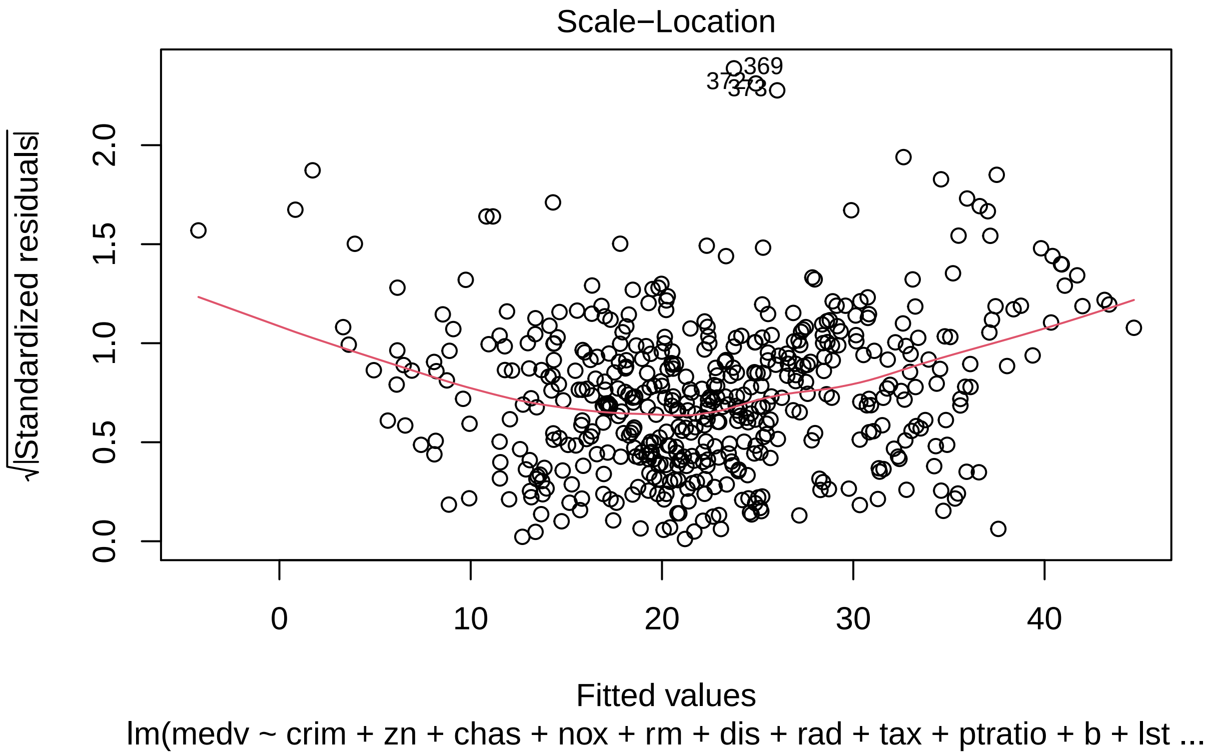
```
summary(final)
```

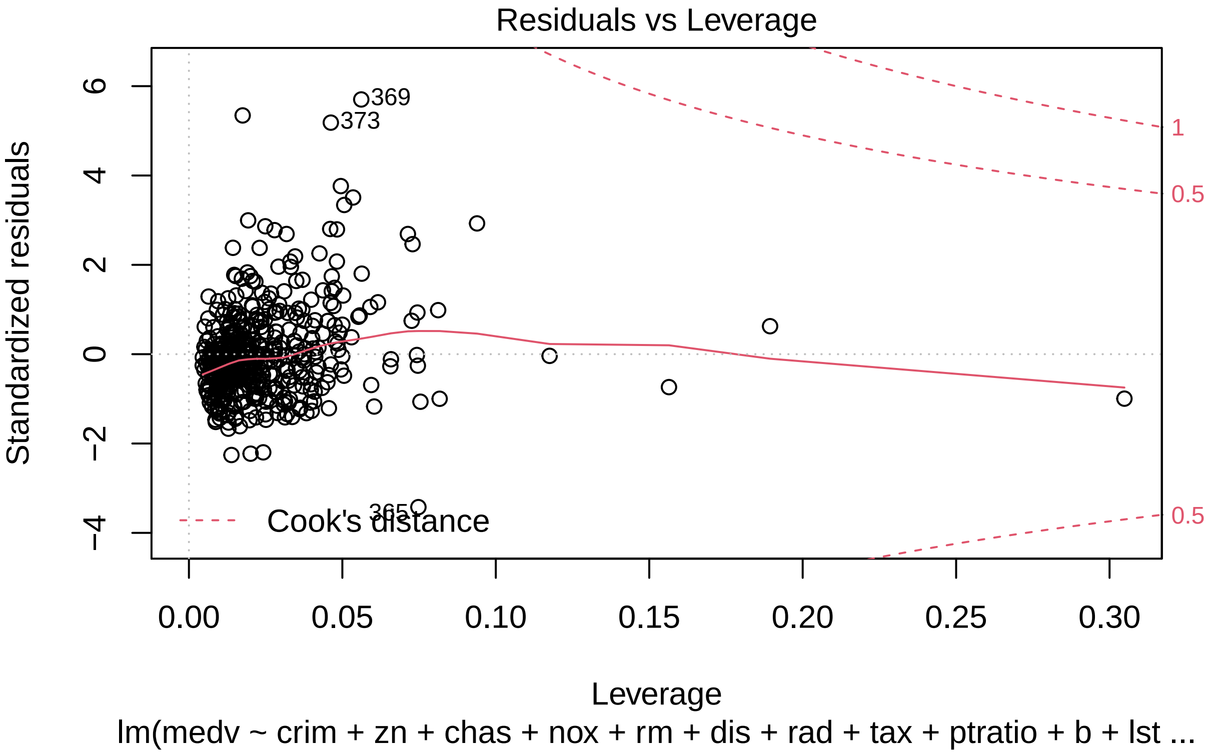
```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + b + lstat, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5984  -2.7386  -0.5046   1.7273  26.2373
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
## crim         -0.108413   0.032779  -3.307 0.001010 **
## zn            0.045845   0.013523   3.390 0.000754 ***
## chas1         2.718716   0.854240   3.183 0.001551 **
## nox         -17.376023   3.535243  -4.915 1.21e-06 ***
## rm            3.801579   0.406316   9.356  < 2e-16 ***
## dis          -1.492711   0.185731  -8.037 6.84e-15 ***
## rad           0.299608   0.063402   4.726 3.00e-06 ***
## tax          -0.011778   0.003372  -3.493 0.000521 ***
## ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
## b             0.009291   0.002674   3.475 0.000557 ***
## lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.736 on 494 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7348
## F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

```
plot(final)
```




Normal Q−Q
Standardized residuals
6
4
2
0
−2
369
372
373
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + b + lst ...


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
369
372
373
0
10
20
30
40
Fitted values
lm(medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + b + lst ...



### Analysis ###

While the plots did not show a significant difference between the base model, the model with non-significant variables manually removed, and the step-AIC optimized model, the adjusted R-squared was higher for the manually adjusted and step-AIC optimized model, thus we can conclude that these models fit the data better. The manually adjusted and step-AIC models appear to be equivalent, so we will use the step-AIC model as our final model.

In the final model, we see that `indus` and `age` do not seem to be significant. The plots for the model reveal a fair model with some room for improvement. The Residuals vs Fitted values plot shows a cluster near the center of the plot, and what appears to be mild decreasing trend for low fitted values and another mild increasing trend for high fitted values. This indicates that the model may have less predictive precision for high or low predictions. The standardized residuals vs Fitted Values plot shows a similar trend, but the clustering near the middle of the plot is less severe. Ideally, we would see no clustering in either plot and no visible trends for high nor low values. Overall however, the non-parametric smoother does not severely deviate from a flat line at zero. The normal Q-Q plot better identifies the flaws of the model. We can clearly see that for very low values, the model seems to overestimate the medv of low predictions. For high theoretical quantities, the errors are more severe, and the model underestimates high medv values with an increasing underestimation bias.

Finally, the residuals vs leverage plot shows that we do not have any outliers which disproportionately influence the model. Thus we can keep all of the data to fit the model.

In conclusion, while the model has some room for improvement at predicting the value of medv, it is accurate enough for us to come to conclusions about which variables are important predictors. We find that:

1. crim is significant, and brings down medv.
2. zn is highly significant, and brings up medv.
3. chas1 is significant, and brings up medv.

4. nox is highly significant and brings down medv.
5. rm is highly significant, brings up medv.
6. dis is highly significant, and brings down medv.
7. rad is highly significant, and brings up medf.
8. Tax is highly significant, and brings down medv.
9. pratio is highly significant, and brings down medv.
10. b is highly significant, and brings up medv.
11. lstat is highly significant, and brings down medv.

Problem 7

```
d <- read.table("T6-8.dat")
head(d)
```

```
##     V1    V2     V3  V4
## 1  869 860.5  691.0 601
## 2  995 875.0  678.0 659
## 3 1056 930.5  833.0 826
## 4 1126 954.0  888.0 728
## 5 1044 909.0  865.0 839
## 6  925 856.5 1059.5 797
```

Part A, repeated measures design

- Construction of contrast matrix, covariance matrix and mean vector

```
C <- rbind(c(1,-1,1,-1),
           c(1,1,-1,-1),
           c(1,-1,-1,1))

x_bar <- colMeans(d)
S <- var(d)
n <- nrow(d)
q <- ncol(d)
```

- Calculation of repeated measures design

```
t_2 <- n*(t(C%*%x_bar)%*%solve(C%*%S%*%t(C)) %*%C%*%x_bar)

#F-statistic
f_stat <- qf(0.05,q-1, n-q-1)
t_2 > f_stat
```

```
##      [,1]
## [1,] TRUE
```

And thus we reject the null hypothesis that C*mu=0 and conclude that we have some treatment effects.

Part B, 95% Confidence Intervals

```
# individual contrasts
c_1 = (C[1,])
c_2 = (C[2,])
```

```
c_3 = (C[3,])


rad_1 <- sqrt( ( (n-1)*(q-1)/(n-q+1) ) * qf(0.05,q-1,n-q+1)  )
rad_2 <- function(c_i) sqrt( t(c_i)%*%S%*%c_i/n )

print("Different vs Same Parity Confidence interval")
```

```
## [1] "Different vs Same Parity Confidence interval"
```

```
upper_1 <- t(c_1)%*%x_bar + rad_1*rad_2(c_1)
lower_1 <- t(c_1)%*%x_bar - rad_1*rad_2(c_1)
c('upper' = upper_1, 'lower' = lower_1)
```

```
##    upper    lower
## 221.4151 191.2412
```

```
print('Word vs Arabic Confidence Interval')
```

```
## [1] "Word vs Arabic Confidence Interval"
```

```
upper_2 <- t(c_2)%*%x_bar + rad_1*rad_2(c_2)
lower_2 <- t(c_2)%*%x_bar - rad_1*rad_2(c_2)
c('upper' = upper_2, 'lower' = lower_2)
```

```
##    upper    lower
## 328.5595 285.2842
```

```
print('Interaction effects')
```

```
## [1] "Interaction effects"
```

```
upper_3 <- t(c_3)%*%x_bar + rad_1*rad_2(c_3)
lower_3 <- t(c_3)%*%x_bar - rad_1*rad_2(c_3)
c('upper' = upper_3, 'lower' = lower_3)
```

```
##     upper     lower
## -11.63532 -33.20843
```

- Different vs Same parity
  - We have a 95% confidence interval of (191.2412 , 221.4151). This confidence interval is far removed from 0 and shows a large effect size for Different vs. Same. It implies that having different parities increases the response time.
- Word vs Arabic confidence interval
  - We have a 95% confidence interval of (285.2842 , 328.5595). This confidence interval is also far removed from 0 and shows an effect due to Word vs Arabic presentation of numbers. Presenting numbers in word format increased the response time
- Interaction in effects
  - We have a 95% confidence interval of (-33.20843 , -11.65532). This confidence interval does not contain 0 and shows an interaction in the treatment effects. The data suggests that presenting numbers in arabic format lessened the response time increase when presented with different parities, or that presenting numbers in word format magnified the response time increase when presented with different parities.

Part C

Since we did observe interaction effects (the 95% confidence interval for interaction effects did not contain 0), the data supports the C and C model of numerical cognition.

Part d

- Generate difference score data matrix and test for normality

```
diff_m <- as.matrix(d) %*% t(C)
colnames(diff_m) = c("parity","format",'interaction')

mvn(diff_m)
```

```
## $multivariateNormality
##              Test         Statistic            p value Result
## 1 Mardia Skewness 19.8907307206908 0.0303032967627702     NO
## 2 Mardia Kurtosis 1.48240196389931  0.138233371738183    YES
## 3             MVN             <NA>               <NA>     NO
##
## $univariateNormality
##           Test    Variable Statistic   p value Normality
## 1 Shapiro-Wilk   parity       0.9348    0.0535    YES
## 2 Shapiro-Wilk   format       0.9586    0.2518    YES
## 3 Shapiro-Wilk interaction    0.9692    0.4763    YES
##
## $Descriptives
##              n      Mean  Std.Dev Median    Min   Max   25th    75th      Skew
## parity      32 206.32812 139.9195 169.25  -34.5 607.0 121.25 283.375 0.8980471
## format      32 306.92188 200.6721 276.75  -75.0 879.5 192.25 438.500 0.6150826
## interaction 32 -22.42188 100.0367 -37.50 -217.0 229.5 -82.50  29.750 0.3086301
##               Kurtosis
## parity       0.6024854
## format       0.6810366
## interaction -0.0977505
```

Based off of the results of the Marida Skewness test, this data is cannot be represented via a multivariate normal model.

Problem 8

- Principle Component Analysis with covariance matrix S

```
d <- read.table("T1-5.dat")
colnames(d) <- c('Wind','Solar_r','CO','NO','NO_2','O_3','HC')
n <- nrow(d)
xbar <- colMeans(d)
S <- var(d)
E <-  eigen(S)


coef_vec <- E$vectors
tot_var <- sum(E$values)
p_var <- E$values
```

```
#data summary
summary(d)
```

```
##       Wind          Solar_r             CO              NO
##  Min.   : 5.00   Min.   : 30.00   Min.   :2.000   Min.   :1.00
##  1st Qu.: 6.00   1st Qu.: 68.25   1st Qu.:4.000   1st Qu.:1.00
##  Median : 8.00   Median : 76.50   Median :4.000   Median :2.00
##  Mean   : 7.50   Mean   : 73.86   Mean   :4.548   Mean   :2.19
##  3rd Qu.: 8.75   3rd Qu.: 84.75   3rd Qu.:5.000   3rd Qu.:3.00
##  Max.   :10.00   Max.   :107.00   Max.   :7.000   Max.   :5.00
##       NO_2            O_3              HC
##  Min.   : 5.00   Min.   : 2.000   Min.   :2.000
##  1st Qu.: 8.00   1st Qu.: 6.000   1st Qu.:3.000
##  Median : 9.50   Median : 8.500   Median :3.000
##  Mean   :10.05   Mean   : 9.405   Mean   :3.095
##  3rd Qu.:12.00   3rd Qu.:11.000   3rd Qu.:3.000
##  Max.   :21.00   Max.   :25.000   Max.   :5.000
```

```
#eigenvalues
p_var
```

```
## [1] 304.2578640  28.2761046  11.4644830   2.5243296   1.2795247   0.5287288
## [7]   0.2096157
```

```
#variance contained in the first principle component
p_var[1]/tot_var
```

```
## [1] 0.872948
```

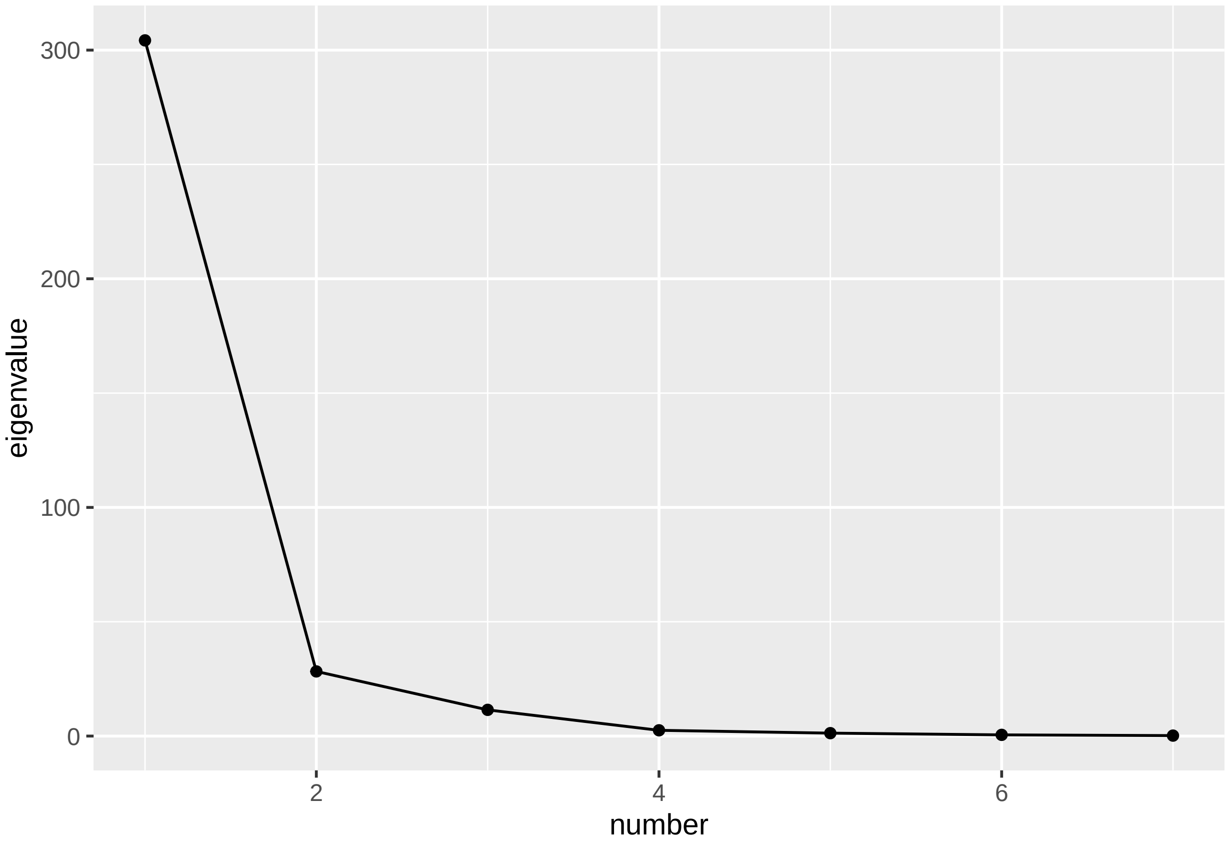
```
#Variance contained in first two principle components
(p_var[1]+p_var[2])/tot_var
```

```
## [1] 0.9540751
```

```
#variance contained in first 3 components
(p_var[1]+p_var[2]+p_var[3])/tot_var
```

```
## [1] 0.986968
```

```
#first 3 components contain 98.7% of the variance

#scree graph further illustrates point
df <- data.frame('eigenvalue'=p_var,'number'= 1:7)
ggplot(data=df,aes(x=number, y=eigenvalue)) + geom_line()+geom_point()
```



```
##correlations between variables and principle components

#obtain diagonal matrix with diagonal entries sigma_ii^(-1/2)
var_ii <- diag(S)
D <- diag(var_ii)^(-1/2)
for(i in 1:nrow(D)){
  for(j in 1:ncol(D))
    if(i != j) {
      D[i,j] = 0
    }
}

#obtain diagonal matrix with entries lambda_i^(1/2)
s_eigen <- sqrt(diag(p_var))

x_cor <- coef_vec  %*% s_eigen
x_cor <- t(t(x_cor) %*% D) # variable correlations with principal components

#Raw Component Vectors:
coef_vec
```

```
##               [,1]        [,2]        [,3]          [,4]          [,5]
## [1,]  0.010039244  0.07622439  0.03087761  0.9203045748  0.3423859285
## [2,] -0.993199405  0.11615518  0.00659069 -0.0002118679  0.0022391022
## [3,] -0.014062314 -0.09956775 -0.18282641 -0.1382922410  0.6500776063
## [4,]  0.004710175  0.01320423 -0.13021553 -0.3277842624  0.6431560485
## [5,] -0.024255644 -0.15038113 -0.95526318  0.1023719020 -0.2065840405
## [6,] -0.112429558 -0.97335904  0.16981025  0.0632480276 -0.0002935726
## [7,] -0.002340785 -0.02382046 -0.08519558  0.1095073458  0.0619613872
##              [,6]         [,7]
## [1,]  0.011779079 -0.169729925
## [2,]  0.003353218 -0.001781987
## [3,] -0.563893916  0.443577538
## [4,]  0.497513370 -0.462855916
## [5,] -0.009009299 -0.105029951
## [6,]  0.051067254 -0.066992404
## [7,]  0.657012233  0.738019426
```

```
#correlation of variables with component vector"
x_cor
```

```
##             [,1]        [,2]         [,3]          [,4]          [,5]
## [1,]  0.11075209  0.25635026  0.066122757  9.247719e-01  2.449459e-01
## [2,] -0.99936420  0.03562992  0.001287285 -1.941801e-05  1.461049e-04
## [3,] -0.19882031 -0.42915242 -0.501763680 -1.780959e-01  5.960361e-01
## [4,]  0.07555890  0.06457294 -0.405478384 -4.789485e-01  6.690651e-01
## [5,] -0.12550964 -0.23721736 -0.959497022  4.824998e-02 -6.932094e-02
## [6,] -0.35234744 -0.92993495  0.103302503  1.805468e-02 -5.966361e-05
## [7,] -0.05902492 -0.18311035 -0.417010749  2.515181e-01  1.013208e-01
##               [,6]          [,7]
## [1,]  0.0054169873 -4.914738e-02
## [2,]  0.0001406516 -4.706335e-05
## [3,] -0.3323509745  1.646131e-01
## [4,]  0.3326970328 -1.948881e-01
## [5,] -0.0019433492 -1.426489e-02
## [6,]  0.0066715801 -5.510705e-03
## [7,]  0.6906259742  4.884641e-01
```

- Principle Component analysis using correlation matrix R

```
R <- cor(d)
E <- eigen(R)
coef_vec <- E$vectors
p_var <- E$values
tot_var <- ncol(R)

#eigenvalues
p_var
```

```
## [1] 2.3367826 1.3860007 1.2040659 0.7270865 0.6534765 0.5366888 0.1558989
```

```
#information weights
p_var/tot_var
```

```
## [1] 0.33382609 0.19800010 0.17200942 0.10386950 0.09335379 0.07666983 0.02227128
```

```
#proportion of variance due to 1st component
p_var[1]/tot_var
```

```
## [1] 0.3338261
```

```
#due to first 2 components
(p_var[1]+p_var[2])/tot_var
```

```
## [1] 0.5318262
```

```
#due to first 3
(p_var[1]+p_var[2]+p_var[3])/tot_var
```

```
## [1] 0.7038356
```

```
#due to first 4
(p_var[1]+p_var[2]+p_var[3]+p_var[4])/tot_var
```

```
## [1] 0.8077051
```

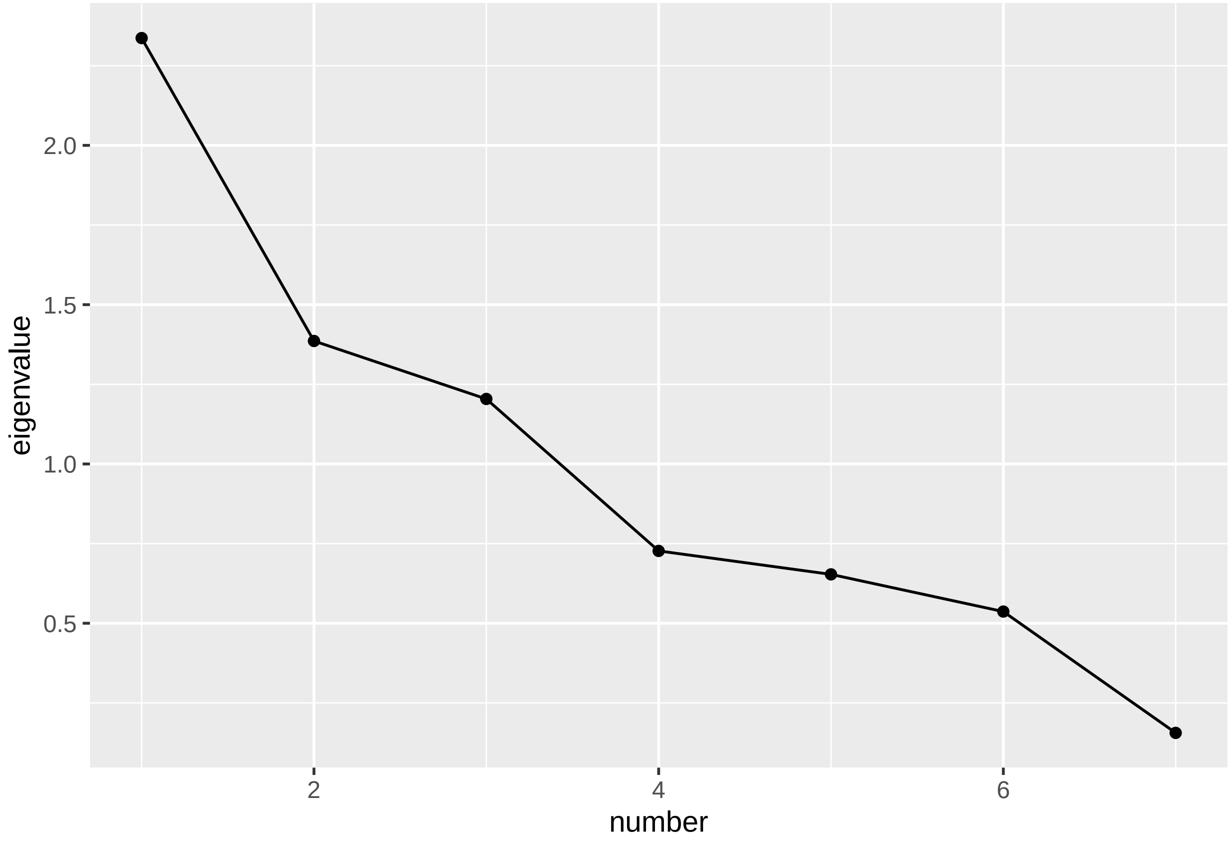
```
#due to first 5
(p_var[1]+p_var[2]+p_var[3]+p_var[4]+p_var[5])/tot_var
```

```
## [1] 0.9010589
```

```
#due to first 6
(p_var[1]+p_var[2]+p_var[3]+p_var[4]+p_var[5]+p_var[6])/tot_var
```

```
## [1] 0.9777287
```

```
# scree plot
df <- data.frame('eigenvalue'=p_var,'number'= 1:7)
ggplot(data=df,aes(x=number, y=eigenvalue)) + geom_line()+geom_point()
```



```
#variable correlations with principle components
```

```
#obtain diagonal matrix with entries lambda_i^(1/2)
s_eigen <- sqrt(diag(p_var))

x_cor <- coef_vec  %*% s_eigen


#Raw Component Vectors:
coef_vec
```

```
##            [,1]         [,2]       [,3]         [,4]        [,5]         [,6]
## [1,]  0.2368211  0.278445138  0.6434744  0.172719491  0.56053441 -0.223579220
## [2,] -0.2055665 -0.526613869  0.2244690  0.778136601 -0.15613432 -0.005700851
## [3,] -0.5510839 -0.006819502 -0.1136089  0.005301798  0.57342221 -0.109538907
## [4,] -0.3776151  0.434674253 -0.4070978  0.290503052 -0.05669070 -0.450234781
## [5,] -0.4980161  0.199767367  0.1965567 -0.042428178  0.05021430  0.744968707
## [6,] -0.3245506 -0.566973655  0.1598465 -0.507915905  0.08024349 -0.330583071
## [7,] -0.3194032  0.307882771  0.5410484 -0.143082348 -0.56607057 -0.266469812
##             [,7]
## [1,] -0.24146701
## [2,] -0.01126548
## [3,]  0.58524622
## [4,] -0.46088973
## [5,] -0.33784371
## [6,] -0.41707805
## [7,]  0.31391372
```

```
#correlation of variables with component vector"
x_cor
```

```
##            [,1]         [,2]       [,3]         [,4]        [,5]         [,6]
## [1,]  0.3620175  0.327809366  0.7060840  0.147276816  0.45312422 -0.163792005
## [2,] -0.3142401 -0.619974764  0.2463097  0.663512149 -0.12621570 -0.004176389
## [3,] -0.8424165 -0.008028499 -0.1246630  0.004520809  0.46354245 -0.080247159
## [4,] -0.5772427  0.511735606 -0.4467081  0.247710112 -0.04582757 -0.329837708
## [5,] -0.7612943  0.235183184  0.2156816 -0.036178238  0.04059219  0.545756973
## [6,] -0.4961256 -0.667489747  0.1753995 -0.433096674  0.06486715 -0.242182006
## [7,] -0.4882569  0.362465859  0.5936921 -0.122005412 -0.45759954 -0.195213244
##              [,7]
## [1,] -0.095340930
## [2,] -0.004448066
## [3,]  0.231078856
## [4,] -0.181977886
## [5,] -0.133394347
## [6,] -0.164679265
## [7,]  0.123945821
```

Analysis

It is certainly possible to summarize the data with 2 dimensions if the principle components are constructed using the sample covariance matrix S. As the Scree plot shows, components after the 2nd component contribute little information to the model. In fact, 95.4% of the variance can be attributed to the first two critical components.

Using the critical components determined by the unormalized data, the component vectors indicate that most of the variance comes from the weighted difference between wind plus NO and all other variables. From the magnitude of the coefficient vector for the first principle component, we see this weighted difference is overwhelmingly determined by Solar_r. This is supported by the correlation between this variable and the first principle component, giving a correlation coefficient of 0.9994 with the first critical component.

The overwhelming influence of Solar_r in determining the variance might be due to the disproportionate size of the Soar_r variables in the data matrix. This indicates that the data may need to be re-normalized to prevent Solar_r from having a disproportionate impact on the variance in the data, which might not reflect the physical model that the data seeks to characterize. However, determining if this is the case likely requires some subject-specific knowledge, so a Data scientist should consult with the research group collecting the data.

Using normalized data and the sample correlation matrix R we have a very different picture of the data. The scree plot shows that all but the last principle component carries a non-insignificant amount of information. We could possibly only use the first 4 components, which carry 81% of the variance. From there the first 5 carry 90% of the variance, and the first 6 97% of the variance. Depending on the subsequent analysis after taking principle components, using the first 4-6 components appears to be appropriate. Using only the first 3 will likely give an incomplete picture of the data (leaves 30% of the variance unaccounted for), and using all 7 is likely excessive, since the 7th principle component only contains 2% of the variance.

With the normalized data Solar radiation plays a much more conservative role in determining the principle components. In this model, the first component appears to be the weighted difference between wind and all other variables in the data. This seems significant to me, since wind would carry away pollutants, whereas all other variables in the data appear to be pollutants. Thus, it would make sense that any model of pollution based off of this data would primarily be determined by the difference between wind, which carries pollution away, and the pollution. Additionally, the variable with the highest “weight”, both in the raw coefficient vector and in the correlation coefficient, is CO, a common and problematic pollutant which is often the consequence of burning fossil fuels. The other principle components are more complicated linear combinations of the variables, and likely need domain-specific knowledge for correct interpretation.

In conclusion, I would argue that the normalized data gives a better picture of the model. The disproportionate impact of Solar Radiation in the unormalized data might be due to a difference in measurement units and might not be reflective of the true physical model under consideration. The linear combinations offered by the normalized data’s principle components also seem to make more intuitive sense in the context of pollution data. Of course, this is all conditional on a true subject-matter expert’s opinion.